Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Please add new claims 44-46.

1. (Currently Amended) A system for message service, comprising:

a business component utilizing messages;

a first queue to manage message services, the first queue employing an architecture other than a publication/subscription type notification;

a wrapper to enable the first queue to operate a publication/subscription notification type of [the] architecture; and

a connector in communication with the first queue via the wrapper, the connector further in communication with the business component, the connector receiving messages from the first queue via the wrapper and ~~communicating~~ sending the messages being received from the first queue via the wrapper to the business component.

2. (Previously Presented) The system of Claim 1, further comprising a second queue to manage the message services, the second queue employing the publication/subscription notification type of architecture and wherein the connector communicates with the second queue to communicate the messages from the second queue to the business component.

3. (Previously Presented) The system of Claim 1, wherein an address identifying a location of at least one of the messages of the first queue is located in a file.

4. (Previously Presented) The system of Claim 1, wherein an address identifying the location of at least one of the messages of the first queue is on a socket connection.

5. (Previously Presented) The system of Claim 1, wherein an address identifying the location of at least one of the messages of the first queue is on a port connection.

6. (Original) The system of Claim 1, wherein the first queue is a polling type queue.

7. (Previously Presented) The system of Claim 2, wherein the connector is further operable to communicate the messages from the business component to at least one of the first and second queues.

8. (Previously Presented) The system of Claim 2, wherein the second queue is further defined as a Java Message Service (JMS) queue.

9. (Previously Presented) The system of Claim 8, wherein the JMS queue receives messages from a file.

10. (Previously Presented) The system of Claim 8, wherein the JMS queue receives messages from a Universal Resource Identifiers (URI) remotely.

11. (Original) The system of Claim 2, wherein the wrapper is further defined as a JMS enabled wrapper.

12. (Original) The system of Claim 2, wherein the second queue is further defined as JMS standards application programming interface (API) operable for inter-client communication.

13. (Original) The system of Claim 2, wherein the publication/subscription notification type of architecture of the first queue enabled by the wrapper facilitates the connector registering with the first queue, via the wrapper, and with the second queue such that when at least one of the first and the second queues receive messages for the connector, the at least one of the first and second queues notify the connector.

14. (Previously Presented) The system of Claim 2, wherein the connector is further defined as a JMS enabled connector.

15. (Original) The system of Claim 2, wherein the connector is operable to register with the wrapper of the first queue as a JMS client.

16. (Previously Presented) A method for processing messages, comprising:

providing a business component;

providing a connector in communication with the business component;

subscribing, by the connector, to a message queue;

providing a message to the message queue, the message being directed for delivery to the business component;

notifying the connector that the message is in the message queue;
obtaining, by the connector, the message from the message queue;
communicating the message to the business component; and
verifying that the business component has received the message.

17. (Original) The method of Claim 16, wherein the connector verifies that the business component has received the message before the message is consumed from the message queue.

18. (Original) The method of Claim 16, wherein the message queue consumes the message.

19. (Original) The method of Claim 16, wherein the connector consumes the message.

20. (Original) The method of Claim 16, wherein the method further includes transforming the message.

21. (Original) The method of Claim 20, wherein transforming the message includes parsing the message and communicating at least a portion of a data portion of the message to the business component.

22. (Original) The method of Claim 16, wherein the method of verifying that the business component received the message includes communicating with the message queue regarding a rate of delivery of the message to the business component.

23. (Previously Presented) The method of Claim 16, wherein the access to the message queue via the connector to the business component includes selectively identifying the message by a portion of the message.

24. (Original) The method of Claim 23, wherein the method further comprises:
prioritizing the message;
transforming the message; and
consuming the message.

25. (Original) The method of Claim 16, wherein the method includes:
providing a second queue utilizing a polling notification type architecture;
providing a wrapper enabling a publication/subscription notification architecture by the second queue; and

registering the connector with the second queue enabling the publication/subscription notification architecture of the wrapper.

26. (Previously Presented) A system of a queue connector to promote message services, comprising:

a first component operable to communicate messages with a publication/subscription notification type queue;

a second component operable to communicate messages with a notification type queue other than the publication/subscription type queue by registering with a wrapper of the publication/subscription notification type queue;

a business component interface operable to communicate with business components; and

a transaction component operable to verify that messages from one of the queues are received by the business components before the messages are consumed, the transaction component deleting a message from one of the queues upon verification of receipt of the message by the business components from the queue from which the message originated.

27. (Previously Presented) The system of Claim 26, further comprising a logging component operable to record information related to the messages including a record of at least some of a message communicated between one of the publication/subscription notification type queue and notification type queue other than the publication/subscription type queue and the business component.

28. (Previously Presented) The system of Claim 27, wherein the record includes a date and time associated with each of the messages.

29. (Previously Presented) The system of Claim 27, wherein the record includes a tracekey associated with each of the messages.

30. (Previously Presented) The system of Claim 29, wherein the tracekey includes information related to the message.

31. (Original) The system of Claim 30, wherein the information included with the tracekey

includes a location of the message.

32. (Original) The system of Claim 30, wherein the information included with the tracekey includes an origin of the message.

33. (Previously Presented) The system of Claim 30, wherein the information included with the tracekey includes a type of the message.

34. (Original) The system of Claim 30, wherein the information included with the tracekey includes a size of the message.

35-38. (Canceled)

40-41. (Canceled)

42. (Previously Presented) The system of claim 26, wherein the first component is a Vitria businessware component.

43. (Previously Presented) The system of claim 26, wherein the notification type queue is an MQ series queue.

44. (New) The system of claim 1, wherein the wrapper is operable to query the first queue to determine if a new message has been received by the first queue.

45. (New) The system of claim 25, wherein the wrapper is operable to query the second queue to determine if a new message has been received by the first queue.

46. (New) The system of claim 26, wherein the wrapper is operable to query the second component to determine if a new message has been received by the second component.